

REMARKS

In the Office Action mailed June 2, 2009 the Office noted that claims 22-41 and 43-46 were pending and rejected claims 22-41 and 43-46. Claims 26 and 30 have been amended, no claim has been canceled, and, thus, in view of the foregoing claims 22-41 and 43-46 remain pending for reconsideration which is requested. No new matter has been added. The Office's rejections and objections are traversed below.

DRAWINGS

The Applicant acknowledges that the drawings submitted on March 16, 2009, were not entered.

OBJECTION TO THE SPECIFICATION

The disclosure stands objected to for informalities. In particular, the Office asserts that the Specification introduces new matter. The Applicant has amended the Specification in compliance with the comments of the Office. The Applicant has further amended claims 26 and 30 consistent with the comments of the Office.

Withdrawal of the objections is respectfully requested.

MEANS PLUS FUNCTION

The Applicant acknowledges that the Office considers claims 34-36, 39, 41, 42, and 44 to satisfy the requirements of

35 U.S.C. § 112, sixth paragraph.

REJECTIONS under 35 U.S.C. § 112

Claims 30, 34-42 and 44 stand rejected under 35 U.S.C. § 112, second paragraph as being indefinite for failing to particularly point out and distinctly claim the subject matter which the applicant regards as the invention. In particular the Office asserts that the claims contain antecedent basis issues and that the means plus function of some of the claims are not supported by the Specification. The Applicant has amended the claims to remove any antecedent basis issues.

The Office asserts that there is no structural support for "means for moving said executable certificate to the host terminal," as in claim 34.

However, ¶ 0027 of the printed publication version of the Specification states

According to one embodiment, the verification apparatus **comprises a smart card or any other secure circuit which is capable of containing, on the one hand, the series of control instructions forming the executable certificate** and, on the other hand, an application effecting the verification test. **The host terminal is provided with a smart card reader** (or with a means for communicating with the secure circuit), and the means for executing the software application to be verified are designed to load and execute in its memory context the series of instructions forming the certificate. The verification application in the smart card or the secure circuit is designed so as to modify the normal running of the execution of the software application to be verified if the result of the execution of the series of control instructions is not transmitted, in conditions previously defined, to the verification application in the smart card or the secure circuit, or

if the result of the verification proves to be negative.

Thus, there is a smart card with the executable certificate and a smart card reader (i.e. the means for moving) on the terminal. The Applicant submits that there is corresponding structure for the feature.

The Office asserts that there is no structural support for "comparison means for positively comparing the result obtained through the execution of the control instructions with the result expected from an authentic application," as in claim 34.

However, ¶ 0011 of the printed publication version of the Specification states

Understood here by the term "positive comparison" is the fact that **any action, operation or modification on the data used by the software application to be verified or any action, operation or modification on the running of the execution of the software application** to be verified produces a behaviour of the software application to be verified identical to that which is expected by the running of the execution of the authentic application. [Emphasis added]

Further ¶ 0036 of the printed publication version of the Specification states

These protected data 2 contain 7 an executable certificate 4 including a series of non-protected control instructions, **which are executed 5 by the application to be verified 1. In practice, the control instructions for the executable certificate 4 are coded in the language of the processor of the host terminal, still called machine language.** As a variant, the instructions for the executable certificate 4 may also be coded in the language of a virtual machine, emulating the behaviour of a processor. [Emphasis added]

It is respectfully submitted that one skilled in the art would understand the program when executed on the host terminal would perform the comparison (i.e. a well know machine level instruction). Thus, the comparison means has structural support in the machine level language performing the comparison.

The Office asserts that there is no structural support for "means which are capable, in the event of a positive comparison, of continuing with the execution of the software application to be verified," as in claim 34.

However, as argued above, the program executing the control instructions and performing comparison to determine further execution is found in the machine language executing the program.

The Office asserts that there is no structural support for "a means for communicating with the secure circuit," as in claim 35.

It is respectfully submitted that the card reader is such a means.

The Office asserts that there is no structural support for "means which is capable of validating or invalidating the authenticity of the software application," as in claim 36.

However, it is respectfully submitted that such authentication is support by the program as argued with reference to ¶¶ 0011 and 0036 as argued above.

The Office asserts that there is no structural support

for "means which are capable of inserting the executable certificate into a first stream of data," as in claim 44.

However, ¶ 0039 states

The **executable certificate** 4 may also be inserted in the **stream of data which the application 1 is capable of processing**. The insertion of executable certificates in a stream of data may correspond to the case where it is necessary to authenticate a protected multimedia stream processing application, accessible to the user on condition that the latter has fulfilled obligations such as defined by the seller of the contents. **The source of the multimedia stream may be a transmission point of a broadcasting network**, the permanent memory of the host terminal, or even a memory unit extractable from the host terminal.

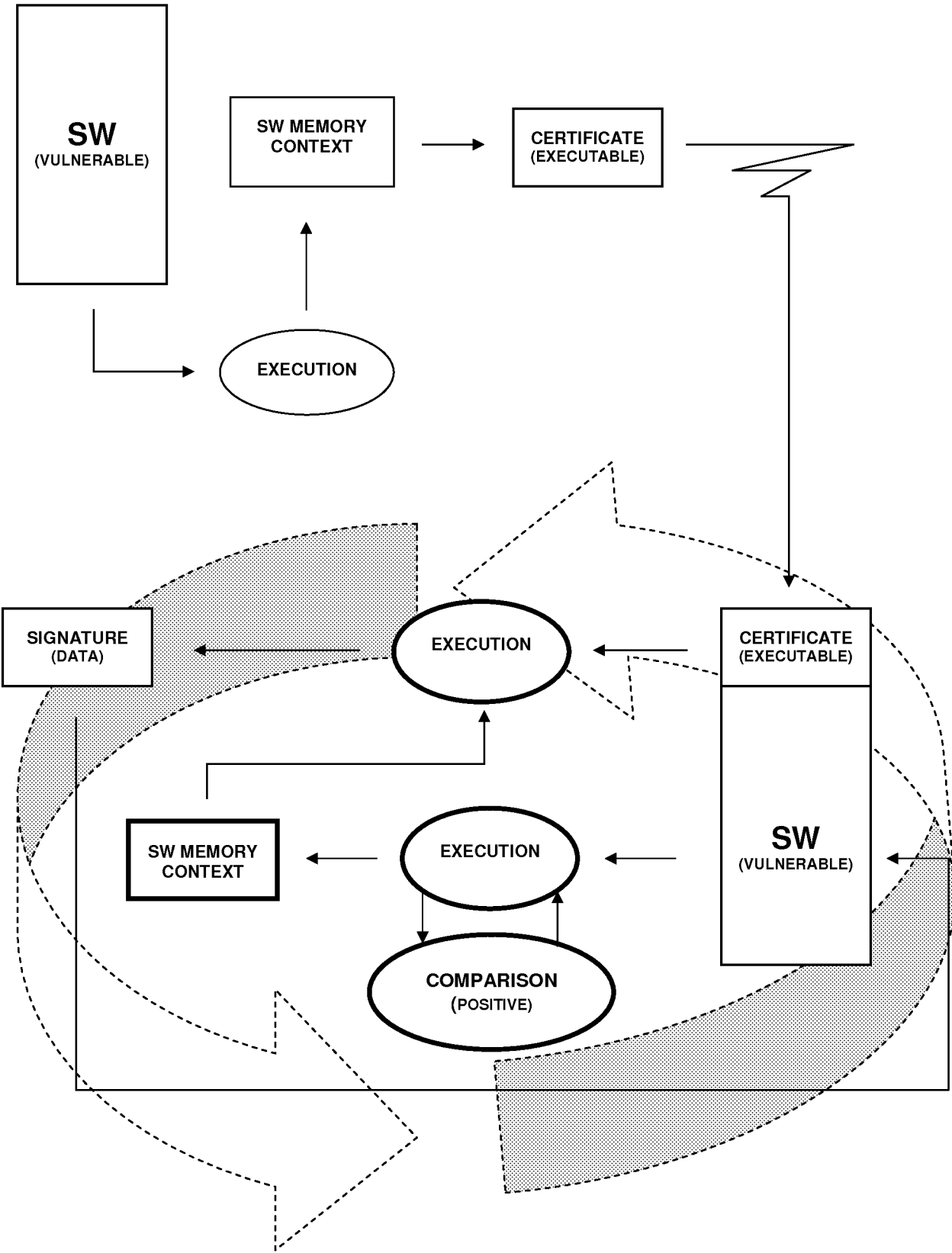
Thus, the executable certificate may be inserted by the transmission point of the broadcasting network. It is respectfully submitted that there is structure for the means which are capable of inserting the executable certificate into a first stream of data.

Withdrawal of the rejections is respectfully requested.

REJECTIONS under 35 U.S.C. § 102

Claims 22-23, 25-41 and 43-46 stand rejected under 35 U.S.C. § 102(e) as being anticipated by McCarroll, U.S. Patent Publication No. 2003/0196102. The Applicant respectfully disagrees and traverses the rejection with an argument.

The Applicant as background for better understanding of the claims submits the following illustration:



It can be distinguished in claim 22 of the application between (a) what is done by the producer of the software (step 1 of the claim, upper part of the illustration) independently of broadcasting the software, and (b) what is done by the host terminal after reception of the executable certificate (steps 2-4 of the claim, lower part of page 2 of the illustration). The software can be loaded in the terminal at any time.

In (a) an authentic software program is run; it has therefore a memory context (software which is not running has no memory context). This memory context is used for determining a series of instructions (an executable certificate i.e. which can be executed on a terminal). And step 1 of the claim also mentions that this series of instructions that can be executed during the execution of the software to be verified, that is later on in (b), on the host terminal.

In contrast with that, McCarroll just digitally signs of portion of the code (See ¶ 0026). Even if this portion of the code is executable, signing it yields a data which obviously is not executable. For doing this, the code only is used, not its memory context. Anyway, the code to be signed is not running and therefore has no memory context. The verification process in McCarroll also excludes that the signature is applied on a portion of the memory context of the running code because the software code signature could not be verified before booting, due

to the fact that before booting, the software to be verified has no memory context.

In (b), step 2, the software, authentic or not, is executed whatever happens; it has therefore, a memory context. The certificate is received and is executed at the same time using the memory context of the running software. To operate, the series of instructions therefore require that the software is running on the terminal. If the software is authentic, its execution combined with the execution of the certificate running at the same time in the software memory context, both produce the expected behavior in the sense that the software does not malfunction. If the software is not authentic, its execution combined with the certificate execution does not match the expected results and the software is therefore unusable. This is the positive comparison (See ¶ 0011). It is not a comparison between two data, but the result of the simultaneous implementation of two processes.

Thus, the executable certificate and the software to be verified are running in the same memory context, then neither can be executed on a separate unit with its own memory context, such as a cryptographic unit which is a secure unit, compulsorily functionally distinct from the host terminal itself.

Further, the executable certificate is generated in (a) after the authentic software is running, the behavior of the software in (b), if not authentic, is not predictable. Even if

the software continues to be executed it does not produces the expected behavior.

In McCarroll ¶ 0029, it is clear that the execution of the software depends on conditions. It is first determined whether the portion of code is valid, as determined by the cryptographic unit. If it is not valid, the system is not booted and therefore the software is not executed. The system is always predictable. The operation of the system which continues as normal in McCarroll (step 214) has nothing to do with an eventual continuing execution of software, which in this case is not yet running. It means that the cryptographic unit is running in a separate memory context than the software to be verified due to the fact that during the verification, the software is not yet booted then it is not yet running.

On page 7 of the Office Action, the Office asserts that processing circuitry and RAM of Fig. 1 (The examiner respectfully points out that the cryptographic unit inherently contains a memory) "using the memory context of the authentic software application during the course of execution," as in claim 22.

The Applicant respectfully disagrees. McCarroll does not disclose a memory context created by the authentic software application. As discussed above, in order for that to occur, the software would have needed to run, which is not the case in McCarroll. In the present claims the memory context of the authentic software is used to produce the certificate in (a), not

to deal with it in (b).

On page 5 of the Office Action, it is asserted that McCarroll ¶¶ 0030-0032 disclose "the software application, which can be executed by said host terminal during the execution of the software application to be verified," as in claim 22.

It is further stated that "Boot process of the disc containing software code for a game." In McCarroll, the host terminal is not executing the software. In fact, McCarroll ¶ 0030 states in part "if the two digests match, it can be verified that the portion of code, such as the Table of Contents, has not been modified since being digitally signed by the manufacturer, and so the boot process is allowed to continue." Thus, in McCarroll, the execution does not occur until after the comparison has been positive, not as part of the execution.

On page 7 of the Office Action, it is asserted that McCarroll, ¶ 0026 disclose "during the course of execution for determining at least one series of control instructions forming an executable certificate for the software application," as in claim 22. It would appear that the Office interprets the "a portion of the software code digitally signed with a key," to be analogous an executable certificate.

However, the claim requires the signed key has at least one series of control instructions. Further, there is no indication that certificate has control instructions. Lastly, there is no requirement of cryptographic method in the present

claims. The two features are therefore, not analogous.

On page 8 of the Office Action, it is asserted that McCarroll Fig. 1 and ¶ 0023 disclose “positively comparing the result thus obtained through the execution of the control instructions in the memory context of said host terminal with the result expected from an authentic software application,” as in claim 22.

However, the comparison (between two behaviors as in the claims) and a mathematical comparison (between two data as in the reference) are not analogous features.

Claim 34 recites similar features as claim 22. Therefore, for at least the reasons argued above, claims 22 and 34 and the claims dependent therefrom are not anticipated by McCarroll.

Withdrawal of the rejections is respectfully requested

REJECTIONS under 35 U.S.C. § 103

Claim 24 stands rejected under 35 U.S.C. § 103(a) as being obvious over McCarroll in view of Yach, U.S. Patent Publication No. 2004/0025022. The Applicant respectfully disagrees and traverses the rejection with an argument.

Yach adds nothing to the deficiencies of McCarroll as applied against the independent claim. Therefore for at least the reasons discussed above, McCarroll and Yach, taken separately or in combination, fail to render obvious the features of claim

24.

SUMMARY

It is submitted that the claims satisfy the requirements of 35 U.S.C. §§ 112, 102 and 103. It is also submitted that claims 22-41 and 43-46 continue to be allowable. It is further submitted that the claims are not taught, disclosed or suggested by the prior art. The claims are therefore in a condition suitable for allowance. An early Notice of Allowance is requested.

The Commissioner is hereby authorized in this, concurrent, and future replies, to charge payment or credit any overpayment to Deposit Account No. 25-0120 for any additional fees required under 37 C.F.R. § 1.16 or under 37 C.F.R. § 1.17.

Respectfully submitted,

YOUNG & THOMPSON

/James J. Livingston/
James J. Livingston, Jr.
Reg.No. 55,394
209 Madison St, Suite 500
Alexandria, VA 22314
Telephone (703) 521-2297
Telefax (703) 685-0573
(703) 979-4709

JJL/fb